PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-168283

(43)Date of publication of application : 13.06.2003

(51)Int.Cl. G11B 27/034
G11B 20/10
G11B 20/12
G11B 27/00
H04N 5/91

(21)Application number : 2001-363588

(22)Date of filing : 29.11.2001

(71)Applicant : SHARP CORP

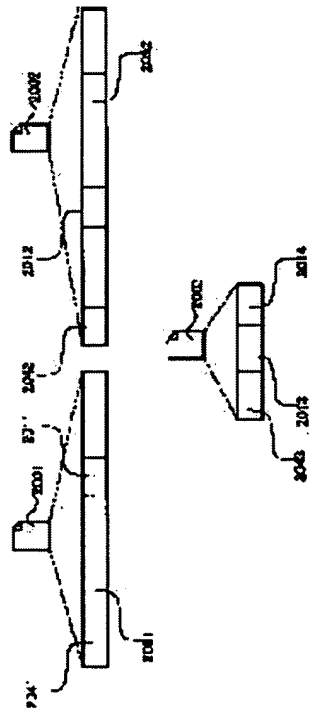
(72)Inventor : KIYAMA JIRO
IWANO HIROTOSHI
YAMAGUCHI TAKAYOSHI

(54) DATA EDITING METHOD AND DATA RECORDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To control reencoded data associated with a nondestructive editing by using a small number of files.

SOLUTION: A data editing method is proposed by which management information which controls a method of reproducing one or more of the first files which store data existing on a recording medium is recorded on the second file and data associated with the existing data are recorded. The associated data and management information are stored in the same file by using the data editing method. The associated data are defined as reencoded data included in the first file, or data reproduced in synchronism with the data included in the first file.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of

rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号
特開2003－168283
（Ｐ２００３－１６８２８３Ａ）
(43)公開日　平成15年６月13日(2003.6.13)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
G11B 27/034		G11B 20/10	G	5C053
20/10		20/12		5D044
20/12		27/00	D	5D110
27/00		27/02	K	
H04N 5/91		H04N 5/91	N	

審査請求　未請求　請求項の数５　ＯＬ　（全19頁）

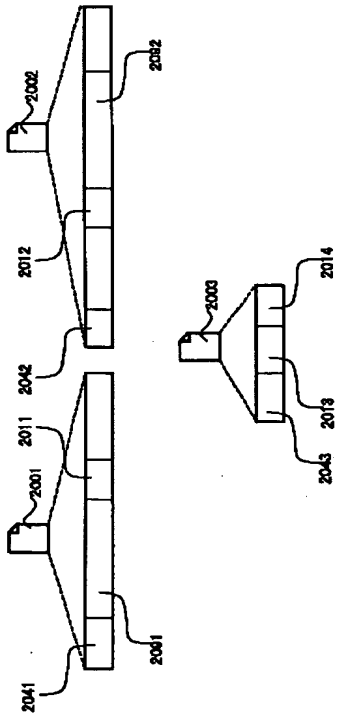
(21)出願番号　特願2001－363588(Ｐ2001－363588)
(22)出願日　平成13年11月29日(2001.11.29)

(71)出願人　000005049
シャープ株式会社
大阪府大阪市阿倍野区長池町22番22号
(72)発明者　木山　次郎
大阪府大阪市阿倍野区長池町22番22号　シャープ株式会社内
(72)発明者　岩野　裕利
大阪府大阪市阿倍野区長池町22番22号　シャープ株式会社内
(74)代理人　100102277
弁理士　佐々木　晴康　（外２名）

最終頁に続く

(54)【発明の名称】データ編集方法およびデータ記録媒体

(57)【要約】
【課題】　少ないファイル数で非破壊編集に伴う再エンコードデータを管理する。
【解決手段】　記録媒体上の既存のデータを格納した1個以上の第1のファイルの再生方法を管理する管理情報を第2のファイルに記録し、前記既存データと関連付けられたデータを記録するデータ編集方法であって、前記関連付けられたデータと前記管理情報とを同一ファイルに格納する。なお、前記関連付けられたデータとは、第1のファイルに含まれるデータを再エンコードしたデータ、或いは、第1のファイルに含まれるデータと同期再生するデータである。



1

【特許請求の範囲】

【請求項１】　記録媒体上の既存のデータを格納した1個以上の第1のファイルの再生方法を管理する管理情報を第2のファイルに記録し、前記既存データと関連付けられたデータを記録するデータ編集方法であって、
前記関連付けられたデータと前記管理情報とを同一ファイルに格納することを特徴とするデータ編集方法。

【請求項２】　記録媒体上の既存のデータを格納した1個以上の第1のファイルの再生方法を管理する管理情報を第2のファイルに記録し、前記既存データと関連付けられたデータを記録するデータ編集方法であって、
前記関連付けられたデータと前記管理情報とを前記記録媒体上の近傍に配置することを特徴とするデータ編集方法。

【請求項３】　前記請求項１又は２に記載のデータ編集方法において、
前記関連付けられたデータは、第1のファイルに含まれるデータを再エンコードしたデータであることを特徴とするデータ編集方法。

【請求項４】　前記請求項１又は２に記載のデータ編集方法において、
前記関連付けられたデータは、第1のファイルに含まれるデータと同期再生するデータであることを特徴とするデータ編集方法。

【請求項５】　データを格納した1個以上の第1のファイルと、前記第1のファイルの再生方法を管理する管理情報を格納する第2のファイルと、前記第1のファイルと関連付けられたデータとが記録されたデータ記録媒体であって、
前記第2のファイル中には前記第1のファイルと関連付けられたデータが格納されていることを特徴とするデータ記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ハードディスク、光ディスク等のランダムアクセス可能な記録媒体に対して、映像データ、音声データを記録・編集するデータ編集方法に関するものである。

【０００２】

【従来の技術】ディスクメディアを用いたビデオのディジタル記録再生装置（以下、ビデオディスクレコーダと呼ぶ）が普及しつつある。テープメディアにはないディスクメディアにおける特徴機能として、非破壊編集機能あるいはノンリニア編集機能と呼ばれるものがある。この機能は、ディスク上に記録したAVストリームを移動あるいはコピーすることなく、AVストリームの任意の区間（シーン）を任意の順番で再生できる、というもので、AVストリームのどこからどこまでどういう順番で再生するかを示す情報（再生管理情報）を作り、その情報に従って再生することで実現される。

2

【０００３】一般的に、ビデオディスクレコーダでのビデオの圧縮にはMPEG-2が用いられる。MPEG-2においては、複数のフレーム（一般的には15フレーム程度）でGOP（Group Of Pictures）が構成され、デコードはGOP単位に行われる。そのため、非破壊編集において、シーン再生開始フレームにGOPの途中のフレームが指定された場合、シーン再生開始フレームの含まれるGOPの先頭からデコードし、シーン再生開始フレーム直前のフレームまでのデコード結果は破棄するように制御する必要がある。

【０００４】この場合、破棄されるフレームのデータもデコーダへ送り、デコードすることになるため、シーン再生開始フレーム付近では単位時間あたりのデコーダへのデータ転送量およびデコーダの処理量が他の個所より高くなり、処理が追いつかず再生に途切れが生じるおそれがある。また、指定されたシーン再生終了フレームがGOPの途中のフレームであった場合、シーン再生終了フレーム以降のデコードを打ち切るように制御する必要がある。つまり、フレーム単位に繋ぎ目が指定された非破壊編集結果を途切れなく再生しようとすると、複雑な制御が要求されることになる。

【０００５】このような問題を解決するための一方法が、特開2001-157145号公報に開示されている。以下、図２９および図３０を用いてその概要を説明する。ここでは、図２９に示すように、ビデオストリームが格納されているファイル3001、3002があったとき、まず、ファイル3001をGOP3011中のフレーム3021まで再生し、次に、ファイル3002をGOP3012中のフレーム3022から再生する、という非破壊編集をすることを想定する。

【０００６】このとき、従来技術では、GOP3011をデコードしフレーム3021までのフレーム列3031と、GOP3012をデコードしフレーム3022からのフレーム列3032とを接続し、フレーム列3033を作り、それをエンコードした結果をファイル3003に格納する。このエンコードのことを再エンコードと呼ぶ。それらの再生順や再生区間を管理するための情報を、図３０に示すようにファイル3003に格納する。

【０００７】再生は、ファイル3003に格納されている情報を基にせず、ファイル3001のGOP3011の直前までのデータ3041、次にファイル3003の全データ3042、最後にファイル3002のGOP3012の直後のデータ3043をデコーダに順に送るだけでよく、デコード結果を破棄したり、デコードの打ち切りのような複雑な制御は必要としない。なぜなら、デコーダに送られるビデオストリームには表示を行うフレームデータしか含まれないようにしているからである。

【０００８】

【発明が解決しようとする課題】しかしながら、上述した従来技術においては、つなぎ目1箇所毎にファイルを作成する必要がある。用いているファイルシステムにお

いては、扱うことができるファイル数が限られている場合があり、その場合、ユーザが作成できる非破壊編集結果の数が少なくなる。

【０００９】本発明は、上記課題を鑑みてなされたものであり、非破壊編集時における再エンコード区間を少ないファイル数で管理することが可能なデータ編集方法を提供することを目的とする。

【００１０】

【課題を解決するための手段】本願の第１の発明は、記録媒体上の既存のデータを格納した1個以上の第1のファイルの再生方法を管理する管理情報を第2のファイルに記録し、前記既存データと関連付けられたデータを記録するデータ編集方法であって、前記関連付けられたデータと前記管理情報とを同一ファイルに格納することを特徴とする。

【００１１】本願の第２の発明は、記録媒体上の既存のデータを格納した1個以上の第1のファイルの再生方法を管理する管理情報を第2のファイルに記録し、前記既存データと関連付けられたデータを記録するデータ編集方法であって、前記関連付けられたデータと前記管理情報とを前記記録媒体上の近傍に配置することを特徴とする。

【００１２】本願の第３の発明は、前記関連付けられたデータが、第1のファイルに含まれるデータを再エンコードしたデータであることを特徴とする。

【００１３】本願の第４の発明は、前記関連付けられたデータは、第1のファイルに含まれるデータと同期再生するデータであることを特徴とする。

【００１４】本願の第５の発明は、データを格納した1個以上の第1のファイルと、前記第1のファイルの再生方法を管理する管理情報を格納する第2のファイルと、前記第1のファイルと関連付けられたデータとが記録されたデータ記録媒体であって、前記第2のファイル中には前記第1のファイルと関連付けられたデータが格納されていることを特徴とする。

【００１５】

【発明の実施の形態】以下、本発明の実施形態について、図面を参照しながら詳細に説明する。ここでの説明は、本発明において共通に用いる構成、個々の実施形態に固有の内容という順に行っていく。

【００１６】＜システム構成＞図１は本発明において共通に用いる、アフレコ可能なビデオディスクレコーダの構成図である。この装置は、図１に示すように、バス100、ホストCPU101、RAM102、ROM103、ユーザインタフェース104、システムクロック105、光ディスク106、ピックアップ107、ECC(Error Correcting Coding)デコーダ108、ECCエンコーダ109、再生用バッファ110、記録/アフレコ用バッファ111、デマルチプレクサ112、マルチプレクサ113、多重化用バッファ114、オーディオデコーダ115、ビデオデコーダ116、オーディオエンコーダ117、ビデオエンコーダ118、および図示しないカメラ、マイク、スピーカ、ディスプレイ等で構成される。

【００１７】ホストCPU101は、バス100を通じてデマルチプレクサ112、マルチプレクサ113、ピックアップ107、また図示していないが、オーディオデコーダ115、ビデオデコーダ116、オーディオエンコーダ117、ビデオエンコーダ118との通信を行う。

【００１８】再生時に、光ディスク106からピックアップ107を通じて読み出されたデータは、ECCデコーダ108によって誤り訂正され、再生用バッファ110に一旦蓄えられる。デマルチプレクサ112はオーディオデコーダ115、ビデオデコーダ116からのデータ送信要求に従い、再生用バッファ中のデータをその種別によって適当なデコーダに振り分ける。

【００１９】一方、記録時に、オーディオエンコーダ117とビデオエンコーダ118によって圧縮符号化されたデータは、多重化用バッファ114に一旦送られ、マルチプレクサ113によってAV多重化され、記録/アフレコ用バッファ111に送られる。記録/アフレコ用バッファ111中のデータは、ECCエンコーダ109によって誤り訂正符号を付加され、ピックアップ107を通じて光ディスク106に記録される。

【００２０】オーディオデータの符号化方式にはMPEG-1 Layer-IIを、ビデオデータの符号化方式にはMPEG-2をそれぞれ用いる。

【００２１】光ディスク106は、外周から内周に向かって螺旋状に記録再生が行われる脱着可能な光ディスクとする。2048byteを1セクタとし、誤り訂正のため16セクタでECCブロックを構成する。ECCブロック中のデータを書き換える場合、そのデータが含まれるECCブロック全体を読み込み、誤り訂正を行って、対象のデータを書き換え、再び誤り訂正符号を付加し、ECCブロックを構成して、記録媒体に記録する必要がある。また、光ディスク106は、記録効率を上げるためZCAV（ゾーン角速度一定）を採用しており、記録領域は回転数の異なる複数のゾーンで構成される。

【００２２】＜ファイルシステム＞光ディスク106上の各種情報を管理するためにファイルシステムを用いる。ファイルシステムには、パーソナルコンピュータ（PC）との相互運用を考慮してUDF（Universal Disk Format）を使用する。ファイルシステム上では、各種管理情報やAVストリームはファイルとして扱われる。

【００２３】ユーザエリアは、2048byteの論理ブロック（セクタと一対一対応）で管理される。各ファイルは、整数個のエクステント（連続した論理ブロック）で構成され、エクステント単位で分散して記録しても良い。空き領域は、Space Bitmapを用いて論理ブロック単位で管理される。

【００２４】＜ファイルフォーマット＞AVストリーム管理のためのフォーマットとして、QuickTimeファイルフ

ォーマットを用いる。QuickTimeファイルフォーマットとは、Apple社が開発したマルチメディアデータ管理用フォーマットであり、PCの世界で広く用いられている。

【００２５】QuickTimeファイルフォーマットは、ビデオデータやオーディオデータ等（これらを総称してメディアデータとも呼ぶ）と管理情報とで構成される。両者を合わせてここでは、QuickTimeムービー（略してムービー）と呼ぶ。両者は同じファイル中に存在しても、別々のファイルに存在しても良い。

【００２６】同じファイル中に存在する場合は、図２（a）に示すような構成をとる。各種情報はatomという共通の構造に格納される。管理情報はMovie atomという構造に格納され、AVストリームはMovie data atomという構造に格納される。尚、Movieatom中の管理情報には、メディアデータ中の任意の時間に対応するAVデータのファイル中での相対位置を導くためのテーブルや、メディアデータの属性情報や、後述する外部参照情報等が含まれている。

【００２７】一方、管理情報とメディアデータを別々のファイルに格納した場合は、図２（b）に示すような構成をとる。管理情報はMovie atomという構造に格納されるが、AVストリームはatomには格納される必要はない。このとき、Movie atomはAVストリームを格納したファイルを「外部参照」している、という。

【００２８】外部参照は、図２（c）に示すように、複数のAVストリームファイルに対して行うことが可能であり、この仕組みにより、AVストリーム自体を物理的に移動することなく、見かけ上編集を行ったように見せる、いわゆる「ノンリニア編集」「非破壊編集」が可能になる。

【００２９】それでは、図３乃至図１２を用いて、QuickTimeの管理情報のフォーマットについて説明する。まず、共通の情報格納フォーマットであるatomについて説明する。atomの先頭には、そのatomのサイズであるAtom size、そのatomの種別情報であるTypeが必ず存在する。Typeは4文字で区別され、例えばMovie atomでは'moov'、Movie data atomでは'mdat'となっている。

【００３０】各atomは別のatomを含むことができる。すなわち、atom間には階層構造がある。Movie atomの構成を図３に示す。Movie header atomは、そのMovie atomが管理するムービーの全体的な属性を管理する。Track atomは、そのムービーに含まれるビデオやオーディオ等のトラックに関する情報を格納する。User data atomは、独自に定義可能なatomである。

【００３１】Track atomの構成を図４に示す。Track header atomは、そのトラックの全体的な属性を管理する。Edit atomは、メディアデータのどの区間を、ムービーのどのタイミングで再生するかを管理する。Track reference atomは、別のトラックとの関係を管理する。Media atomは、実際のビデオやオーディオといったデータを管理する。

【００３２】Track header atomの構成を図５に示す。ここでは、後での説明に必要なもののみについて説明する。flagsは属性を示すフラグの集合である。代表的なものとして、Track enabledフラグがあり、このフラグが1であれば、そのトラックは再生され、0であれば再生されない。layerはそのトラックの空間的な優先度を表しており、画像を表示するトラックが複数あれば、layerの値が小さいトラックほど画像が前面に表示される。

【００３３】Media atomの構成を図６に示す。Media header atomは、そのMedia atomの管理するメディアデータに関する全体的な属性等を管理する。Handler referenceatomは、メディアデータをどのデコーダでデコードするかを示す情報を格納する。Media information atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。

【００３４】Media information atomの構成を図７に示す。Media information header atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。Handler reference atomは、Media atomの項で説明した通りである。Data information atomは、そのQuickTimeムービーが参照するメディアデータを含むファイルの名前を管理するatomであるData reference atomを含む。Sample table atomは、データのサイズや再生時間等を管理している。

【００３５】次に、Sample table atomについて説明するが、その前に、QuickTimeにおけるデータの管理方法について、図８を用いて説明する。QuickTimeでは、データの最小単位（例えばビデオフレーム）をサンプルと呼ぶ。個々のトラック毎に、サンプルには再生時間順に1から番号（サンプル番号）がついている。

【００３６】また、QuickTimeフォーマットでは、個々のサンプルの再生時間長およびデータサイズを管理している。また、同一トラックに属するサンプルが再生時間順にファイル中で連続的に配置された領域をチャンクと呼ぶ。チャンクにも再生時間順に、1から番号がついている。

【００３７】さらに、QuickTimeフォーマットでは、個々のチャンクのファイル先頭からのアドレスおよび個々のチャンクが含むサンプル数を管理している。これらの情報に基づき、任意の時間に対応するサンプルの位置を求めることが可能となっている。

【００３８】Sample table atomの構成を図９に示す。Sample description atomは、個々のチャンクのデータフォーマット（Data format）やサンプルが格納されているファイルのチャンクの Index等を管理する。Time-to-sample atomは、個々のサンプルの再生時間を管理する。

【００３９】Sync sample atomは、個々のサンプルのうち、デコード開始可能なサンプルを管理する。Sample-t

o-chunk atomは、個々のチャンクに含まれるサンプル数を管理する。Sample size atomは、個々のサンプルのサイズを管理する。Chunk offset atomは、個々のチャンクのファイル先頭からのアドレスを管理する。

【００４０】Edit atomは、図１０に示すように、1個のEdit list atomを含む。Edit listatomはNumber of entriesで指定される個数分の、Track duration、Media time、Media rateの値の組（エントリ）を持つ。各エントリは、トラック上で連続的に再生される区間に対応し、そのトラック上での再生時間順に並んでいる。

【００４１】Track durationはそのエントリが管理する区間のトラック上での再生時間、Media timeはそのエントリが管理する区間の先頭に対応するメディアデータ上での位置、Media rateはそのエントリが管理する区間の再生スピードを表す。尚、Media timeが-1の場合は、そのエントリのTrack duration分、そのトラックでのサンプルの再生を停止する。この区間のことをempty editと呼ぶ。

【００４２】図１１にEdit listの使用例を示す。ここでは、Edit list atomの内容が図１１（a）に示す内容であり、さらにサンプルの構成が図１１（b）であったとする。尚、ここではi番目のエントリのTrack durationをD(i)、Media timeをT(i)、Media rateをR(i)とする。このとき、実際のサンプルの再生は、図１１（c）に示す順に行われる。このことについて簡単に説明する。

【００４３】まず、エントリ#1はTrack durationが13000、Media timeが20000、Media rateが1であるため、そのトラックの先頭から13000の区間はサンプル中の時刻20000から33000の区間を再生する。次に、エントリ#2はTrack durationが5000、Mediatimeが-1であるため、トラック中の時刻13000から18000の区間、何も再生を行わない。

【００４４】最後に、エントリ#3はTrack durationが10000、Media timeが0、Media rateが1であるため、トラック中の時刻18000から28000の区間において、サンプル中の時刻0から10000の区間を再生する。

【００４５】図１２にUser data atomの構成を示す。このatomには、QuickTimeフォーマットで定義されてない独自の情報を任意個数格納することができる。1個の独自情報は1個のエントリで管理され、1個のエントリはSizeとTypeとUser dataで構成される。Sizeはそのエントリ自体のサイズを表し、Typeは独自情報をそれぞれ区別するための識別情報、User dataは実際のデータを表す。

【００４６】＜AVストリームの形態＞まず、本実施例におけるAVストリームの構成について、図１３及び図１４を用いて説明する。AVストリームは整数個のRecord Unit（RU）で構成される。RUはディスク上で連続的に記録する単位である。RUの長さは、AVストリームを構成するRUをどのようにディスク上に配置してもシームレス再生（再生中に画像や音声が途切れないで再生できること）やリアルタイムアフレコ（アフレコ対象のビデオをシームレス再生しながらオーディオを記録すること）が保証されるように設定される。この設定方法については後述する。

【００４７】また、RU境界がECCブロック境界と一致するようにストリームを構成する。RUのこれらの性質によって、AVストリームをディスクに記録した後も、シームレス再生を保証したまま、ディスク上でRU単位の配置を容易に変更することができる。

【００４８】RUは、整数個のVideo Unit（VU）で構成される。VUは単独再生可能な単位であり、そのことから再生の際のエントリ・ポイントとなりうる。

【００４９】VU構成を図１４に示す。VUは、1秒程度のビデオデータを格納した整数個のGOP（グループ・オブ・ピクチャ）と、それらと同じ時間に再生されるメインオーディオデータを格納した整数個のAAU（オーディオ・アクセス・ユニット）とから構成される。

【００５０】尚、GOPは、MPEG-2ビデオ規格における画像圧縮の単位であり、複数のビデオフレーム（典型的には15フレーム程度）で構成される。AAUはMPEG-1 Layer-II規格における音声圧縮の単位で、1152点の音波形サンプル点により構成される。サンプリング周波数が48kHzの場合、AAUあたりの再生時間は0.024秒となる。VU中では、AV同期再生のために必要となる遅延を小さくするため、AAU、GOPの順に配置する。

【００５１】また、VU単位で独立再生を可能とするために、VU中のビデオデータの先頭にはSequence Header（SH）を置く。VUの再生時間は、VUに含まれるビデオフレーム数にビデオフレーム周期をかけたものと定義する。さらに、VUを整数個組み合わせてRUを構成する場合、RUの始終端をECCブロック境界に合わせるため、VUの末尾を0で埋める。

【００５２】＜AVストリーム管理方法＞AVストリームの管理方法は、前述のQuickTimeファイルフォーマットをベースにしている。図１５にAVストリーム管理形態を示す。ビデオトラックは、各ビデオフレームを1サンプル（ビデオサンプル）、VU中のビデオの塊を1チャンク（ビデオチャンク）として管理する。メインオーディオトラックは、AAUを1サンプル（オーディオサンプル）、VU中のオーディオの塊を1チャンク（オーディオチャンク）として管理する。

【００５３】＜RU単位決定方法＞次に、RU単位決定方法について説明する。この決定方法では、基準となるデバイス(リファレンス・デバイス・モデル)を想定し、その上でシームレス再生が破綻しないように連続記録単位を決める。

【００５４】それではまず、リファレンス・デバイス・モデルについて、図１６を用いて説明する。リファレン

ス・デバイス・モデルは１個のピックアップとそれにつながるECCエンコーダ・デコーダ501、トラックバッファ502、デマルチプレクサ503、アフレコ用バッファ504、オーディオエンコーダ509、ビデオバッファ505、オーディオバッファ506、ビデオデコーダ507、オーディオデコーダ508とによって構成される。

【００５５】本モデルにおけるシームレス再生は、VUのデコード開始時にトラックバッファ502上に少なくとも1個VUが存在すれば保証されるものとする。オーディオフレームデータのECCエンコーダ501へのデータの入力速度およびECCデコーダ501からデータの出力速度はRsとする。

【００５６】また、アクセスによる読み出し、記録の停止する最大期間をTaとする。さらに、短いアクセス(100トラック程度)に要する時間をTkとする。なお、これら期間には、シーク時間、回転待ち時間、アクセス後に最初にディスクから読み出したデータがECCから出力されるまでの時間が含まれる。本実施例では、Rs=20Mbps、Ta=1秒、Tk=0.2秒とする。

【００５７】前記リファレンス・デバイス・モデルにおいて再生を行った場合、次のような条件を満たせば、トラックバッファ502のアンダーフローがないことが保証できる。

【００５８】条件を示す前にまず、記号の定義を行う。AVストリームを構成するi番目の連続領域をC#iとし、C#i中に含まれる再生時間をTc(i)とする。Tc(i)はC#i中に先頭が含まれているVUの再生時間の合計とする。また、C#iからC#i+1へのアクセス時間をTaとする。

【００５９】また、再生時間Tc(i)分のVU読み出し時間をTr(i)とする。このとき、トラックバッファ502をアンダーフローさせない条件とは、分断ジャンプを含めた最大読み出し時間をTr(i)としたとき、任意のC#iにおいて、

Tc(i)≧Tr(i)+Ta・・・＜式１＞

が成立することである。

【００６０】なぜなら、この式は、シームレス再生の十分条件である、

【００６１】

【数１】

$$\sum_{i} Tc(i) \geq \sum_{i} (Tr(i) + Ta)$$

【００６２】を満たす十分条件であるためである。

【００６３】＜式1＞中のTr(i)に、Tr(i)=Tc(i)×(Rv+Ra)/Rsを代入して、Tc(i)で解くとシームレス再生を保証可能なTc(i)の条件

Tc(i)≧(Ta×Rs)/(Rs-Rv-Ra)・・・＜式２＞

が得られる。

【００６４】つまり、各連続領域に先頭の含まれるVUの合計が上式を満たすようにすれば、シームレス再生を保証可能である。このとき、各連続領域には合計の再生時間が上式を満たす完全なVU群を含むように制限しても良い。

【００６５】自動分割ムービーファイルでも＜式２＞を満たす必要がある。ただし、先頭の自動分割ムービーの最初のRUおよび末尾の自動分割ムービーの最後のRUは＜式２＞を満たさなくてもよい。なぜなら、先頭は記録媒体からのデータ読み出し開始より再生開始を遅らせることにより吸収でき、末尾については次に続くデータがないため、連続再生を気にする必要が無いからである。このように先頭と末尾において条件を緩めることにより、短い空き領域を有効利用できる。

【００６６】＜インデックス・ファイル＞ディスク内に含まれるQuickTimeムービーや静止画データ等を含む各種コンテンツ（以後、AVファイルと呼ぶ）を管理するため、AV Indexファイルという特別のQuickTimeムービーファイルをディスク内に１個置く。

【００６７】図１７に、AV Indexファイルの構成を示す。AV Indexファイルは通常のQuickTimeムービーファイルと同様、管理情報であるMovie atom1791とデータ自体のMovie data atom1792で構成される。AV Indexファイルは、複数のエントリを管理し、ディスク内の各AVファイルはそれぞれ1個のエントリで管理される。さらに、各AVファイルをまとめるための入れ物（以後フォルダと呼ぶ）等もそれぞれ1個のエントリで管理する。

【００６８】Movie atom1791は、各エントリの属性情報を管理するためのProperty track1793、各エントリのタイトル文字列データを管理するためのTitle track1794、各エントリのサムネイル画像データを管理するためのThumbnail track1795、各エントリの代表オーディオデータを管理するためのIntro music track1796の計4種類のトラックで構成される。

【００６９】各エントリに関する属性情報は、それぞれの1792～1795のトラックのサンプルとして管理される。例えばAVファイル1740に関する属性情報はProperty track1793上のサンプル1701、タイトル文字列データはTitle track1794上のサンプル1711、サムネイル画像データはThumbnail track1795上のサンプル1721、代表オーディオデータはIntro music track1796上のサンプル1731で管理する。サンプル間の対応付けは、各サンプルの再生開始時間に基づき行う。すなわち、トラック間で同一時刻に位置するサンプルが同一エントリに対応していると判断する。

【００７０】Movie data atom1793は、各AVファイルに関する属性情報や、タイトル文字列データ、サムネイル画像データ、代表オーディオデータを格納する。属性情報は図１８に示す構成を取る。各フィールドについて説明する。versionは、ファイルフォーマットのバージョンを示す。pe-flagsは各種フラグをまとめたものであり、詳細は後述する。

【００７１】parent-entry-numberは、属性情報に対応

するエントリが属するフォルダに対応するエントリのentry-numberを格納、entry-numberは、属性情報に対応するエントリのentry-numberを格納する。この2個の情報で、ファイルとフォルダの包含関係を表す。set-dependent-flagsおよびuser-private-flagsについては、説明を省略する。

【００７２】creation-timeおよびmodification-timeはこの属性情報に対応するエントリが作成された日時、修正された日時を表す。durationはこの属性情報に対応するエントリの再生時間を表す。binary-file-identifierは、この管理情報に対応するエントリがファイルに対応していた場合、そのファイルのパス名を固定長にエンコーディングしたもので、詳細についての説明は省略する。

【００７３】referred-counterはこの属性情報に対応するエントリが管理するファイルが他のファイルから参照されている回数を格納する。referring file listは、実際に参照しているファイルのパス名のリストを格納する。URL file identifierは、管理するファイルが上記のbinary-file-identifierにエンコードできない場合にURL（Unified Resource Locator）形式で、ファイルのパスを格納する。

【００７４】＜第１の実施形態＞本発明の第１の実施形態について、非破壊編集を行う場合の処理について図１９から図２６を用いて説明する。ここでは、図１９に示すように、AVストリームが格納されているQuickTimeファイル2001、2002があったとき、まず、ファイル2001をVU2011中のGOP列2051中のフレーム2021まで再生し、次に、ファイル2002をVU2012中のGOP列2052中のフレーム2022から再生する、というGOPの途中で繋ぐ非破壊編集をすることを想定する。なお、Movie atom2041、2042はそれぞれ、ファイル2001、2002を再生するための管理情報である。

【００７５】以下では、まずAVストリームに関する処理について説明し、次に管理情報に関する処理について説明する。

【００７６】＜非破壊編集処理：AVストリームに関する処理＞非破壊編集時のAVストリームに関する処理について、図２０を用いて説明する。前述のVU2011からGOP列2051、VU2012からGOP列2052を抜き出し、ビデオデコーダ116でそれぞれビデオフレーム列2031、2032にデコードする。

【００７７】次に、デコードされたビデオフレーム列2031中のビデオフレーム2021までの部分ビデオフレーム列2033からビデオエンコーダ116でエンコード（再エンコード）し、GOP列2053を作成し、部分ビデオフレーム列2033に時間的に対応する部分AAU列2071と結合することでVU2013を作成する。

【００７８】同時に、デコードされたビデオフレーム列2032中のビデオフレーム2022以降の部分ビデオフレーム列2034に関してはビデオエンコーダ116でエンコード（再エンコード）し、GOP列2054を作成し、部分ビデオフレーム列2034に時間的に対応する部分AAU列2072と結合することでVU2014を作成する。すなわち2個のVUを作成する。

【００７９】このとき、2個のVUではなく1個のVUにまとめることも考えられるが、以下に説明する理由により、2個のVUで構成することにする。1個のVUにまとめる場合の手順を図２１に示す。部分ビデオフレーム列2033と2034を結合し、ビデオフレーム列2035を作成し、それをエンコードした結果のGOP列2055と、部分AAU列2071と2072を繋げた結果であるAAU列2075を結合することでVU2015を作成する。

【００８０】この場合の問題点として、つなぎ目において、オーディオ、ビデオのいずれかに時間的隙間が発生する。なぜなら、図２２に示すように、ビデオフレーム周期2081とAAU周期2082が整数倍の関係になっていないため、部分ビデオフレーム列2083と部分ビデオフレーム列2084を時間的隙間無く再生しようとすると、部分AAU列2071と2072のつなぎ目において隙間2083ができることになる。

【００８１】2個のVUの場合でもこの隙間2083はできるが、1個のVUの場合、VUの途中に隙間2083が発生するため、VU単位に移動を行う場合を考えた場合、処理が複雑化する可能性がある。そのため2個のVUで構成することにする。

【００８２】＜非破壊編集処理：管理情報に関する処理＞非破壊編集時の管理情報に関する処理について、図２３を用いて説明する。まず、ファイル2001のMovie atom2041からVU2011直前までのVU列に関する情報を取得する。同様に、ファイル2002のMovie atom2042 のSample table atomから、VU2012直後より後のVU列に関する情報を取得する。

【００８３】次に、新規に作成した前述のVU2013とVU2014に関して、Sample table作成に必要な情報、具体的にはビデオチャンク、オーディオチャンクのデータサイズおよび再生時間等を取得する。それらの情報を元に非破壊編集結果に関するSample table atomをビデオトラック、オーディオトラックそれぞれについてRAM102上で再構築する。

【００８４】次に、前記Sample table atom中のサンプルを順に隙間無く再生するようにEdit list atomを構成する。ただし、VU2013とVU2014のつなぎ目に関しては、VU2014以降が正しくAV同期再生可能なように、前述のようにオーディオトラックに関して、無再生区間を挿入する必要がある。これらの情報を基に、非破壊編集結果に関するMovie atom2043を作成し、VU2013およびVU2014と共にファイル2003にまとめる。ファイル2003を記録する際には、記録媒体上で連続的に配置されるように記録する。

【００８５】このように非破壊編集に伴い再エンコードしたデータと非破壊編集結果に関する管理情報を1個のファイルにまとめることによって、次のようなメリットが生じる。まず、ファイル数の増加が抑えられる点である。ファイルシステムによっては管理可能なファイル数の上限が存在するため、そのようなファイルシステムにおいてはファイル数が少なくて済むということは、より多数のコンテンツを記録できることを意味する。なお、本実施形態ではファイル2003を記録する際、連続的に記録しているが、連続的でなくてもファイル数増加抑制の効果があることは言うまでもない。

【００８６】なお、本実施形態では、つなぎ目を含むVU（VU2011とVU2012）のみを抜き出して１個のファイル2003にまとめたが、非破壊編集結果がシームレス再生可能なよう、ファイル2003中のAVデータが前述の＜式２＞を満たすようにつなぎ目を含むVU以外のVUも抜き出すことも考えられる。

【００８７】図２４を用いて説明する。VU2011はRU2101に含まれ、VU2012はRU2102に含まれるとする。このとき、RU2101に含まれるVU2011直前のVU列2103と再エンコードしたVU2013とVU2014とRU2102に含まれるVU2012より後のVU列2112の合計の再生時間が＜式２＞を満たすのであれば、ファイル2003にVU列2103、VU2013、VU2014、VU列2112を記録媒体上で連続に格納する。仮に＜式２＞に満たないのであれば、RU2101の直前のRUであるRU2103もコピーする。このことにより、ファイル2003が＜式２＞を満たし、非破壊編集結果に関してもシームレス再生を保証することが可能になる。

【００８８】＜再生時の処理＞非破壊編集結果ファイル2003の再生が指示された場合、まず、Movie atom2043を光ディスク106からRAM102上に読み出す。読み出したMovie atomの情報に基づき、再生順に、VU列2091、VU2013、VU2014、VU列2092の順（図２５中の(1)～(5)の順）に光ディスク106から再生用バッファ110に読み出していく。

【００８９】再生用バッファ110に読み出されたAVデータは、Movie atomの情報に基づき、デマルチプレクサ112によって、オーディオデータとビデオデータに分けられ、それぞれオーディオデコーダ115、ビデオデコーダ116に送られる。オーディオデコーダ115、ビデオデコーダ116はMovie atomの情報に基づくホストCPU101からの制御により、同期を取って再生を行う。

【００９０】このとき、図２６の（1）～（5）に示す順番で読み出しを行う、すなわち、VU2013、VU2014、VU列2091、VU列2092の順に行うことも考えられる。VU2013およびVU2014は実際にはVU列2091の後に再生されるため、VU列2091の再生が終わるまで再生用バッファ110にVU2013およびVU2014から読み出したデータを保持しておき、VU列2091の再生が終了すると同時に、再生を行う。

【００９１】このようにすることで、再生順に読み出した場合と比較して、シーク回数が1回減るため、シークに伴うモーターの消費電力が削減できる。非破壊編集結果の管理情報と再エンコードデータをディスク上で物理的に連続的に記録しておくことで、上述の効果を実現可能である。なお、非破壊編集結果の管理情報と再エンコードデータは連続してなくとも近傍に配置されていれば同様の効果を実現可能なのは言うまでもない。

【００９２】＜第２の実施形態＞本実施形態では、オリジナルデータファイルに影響を与えず、部分的な区間にエフェクトをかける処理について、図２７を用いて説明する。ここでは、オリジナルデータがファイル2201に格納されており、区間2221に対してエフェクト（例えばモザイク）をかけることを想定する。

【００９３】このとき、区間2221を含むRU列2232中のGOP列をデコードし、生成された非圧縮ビデオフレームデータに対し、指定区間に対しエフェクトを施し、エフェクト結果に対してエンコードを行い、RU列2234を再構成する。RU列2234はMovie atom2212と共に1個のファイル2202に格納する。

【００９４】Movie atom2212には、RU列2231、RU列2234、RU列2233の順に再生するように管理情報を格納する。なおRU列2234は記録媒体上で連続的に配置されるよう記録する。

【００９５】以上の構成により、上述した第１の実施形態と同様、ファイルを1個増やすのみでオリジナルデータファイルに影響を与えることなく、しかもオリジナルデータをすべてコピーすることなく部分的な区間に対しエフェクトをかけることが可能となる。また、ファイル2202に格納するデータはRUを考慮しているため、エフェクト結果もシームレス再生を保証することが可能である。なお、本実施形態ではＲＵ列2234を記録する際、連続的に記録しているが、連続的でなくてもファイル数増加抑制の効果があることは言うまでもない。

【００９６】＜第３の実施形態＞本実施形態では、オリジナルデータに影響を与えず、オーディオアフターレコーディング（アフレコ）を行う処理について、図２８を用いて説明する。ここでは、オリジナルデータ2321がファイル2301に格納されており、任意の区間に対してアフレコを行うことを想定する。

【００９７】このとき、アフレコ時に入力されたオーディオデータ2322をファイル2302に、Movie atom2312と共に格納する。Movie atom2312には、オリジナルデータ2321とオーディオデータ2322を同期再生するように管理情報を格納する。Movie atom2312とオーディオデータ2322は記録媒体上で連続的に配置されるように記録する。

【００９８】以上の構成により、上述した第１の実施形態と同様、ファイルを1個増やすのみでオリジナルデータに影響を与えることなく、アフレコデータを管理することが可能となる。なお、本実施形態ではMovie atom2312とオーディオデータ2322を記録する際、連続的に記録

しているが、連続的でなくてもファイル数増加抑制の効果があることは言うまでもない。

【００９９】また、Movie atom2312とオーディオデータ2322は記録媒体上で連続的に配置されていることにより、Movie atom2312とオーディオデータ2322をシークすることなく連続的に読み込むことができるため、ユーザからの再生指示から実際に再生が開始されるまでの間、時間を要することなく、さらに再生中にオーディオデータ2322へのシークを行うことなく、オーディオデータ2322とオリジナルデータ2321との同期再生が可能となる。

【０１００】

【発明の効果】以上説明したように、本発明によれば、非破壊編集時に再エンコードしたデータを非破壊編集結果の管理情報と同じファイルに格納することで、オリジナルデータファイルを書きかえず、しかもファイル数の増加を1個のみに抑えることが可能となる。

【０１０１】また、本発明によれば、非破壊編集時に再エンコードしたデータを非破壊開編集結果の管理情報と記録媒体上で近傍に記録することで、再生時のアクセスを減少することが可能になる。

【図面の簡単な説明】

【図１】本発明の実施形態における概略構成を示すブロック図である。

【図２】QuickTimeファイルフォーマットにおける管理情報とAVストリームとの関係を示す説明図である。

【図３】QuickTimeファイルフォーマットにおけるMovie atomの概要を示す説明図である。

【図４】QuickTimeファイルフォーマットにおけるTrack atomの概要を示す説明図である。

【図５】QuickTimeファイルフォーマットにおけるTrack header atomの構成を示す説明図である。

【図６】QuickTimeファイルフォーマットにおけるMedia atomの構成を示す説明図である。

【図７】QuickTimeファイルフォーマットにおけるMedia information atomの構成を示す説明図である。

【図８】Sample table atomによるデータ管理の例を示す説明図である。

【図９】QuickTimeファイルフォーマットにおけるSample table atomの構成を示す説明図である。

【図１０】QuickTimeファイルフォーマットにおけるEdit atomの構成を示す説明図である。

【図１１】Edit atomによる再生範囲指定の例を示す説明図である。

【図１２】QuickTimeファイルフォーマットにおけるUser data atomの構成を示す説明図である。

【図１３】本発明におけるAVストリームの構成を示す説明図である。

【図１４】本発明におけるVUの構造を示す説明図である。

【図１５】本発明におけるAVストリーム管理形態を示す説明図である。

【図１６】本発明におけるリファレンス・デバイス・モデルを示す説明図である。

【図１７】本発明におけるAV Indexの構成を示す説明図である。

【図１８】本発明におけるAV Index中の属性情報の構成を示す説明図である。

【図１９】本発明の第１の実施形態における、非破壊編集の条件を示す説明図である。

【図２０】本発明の第１の実施形態における、第1の再エンコード方法を示す説明図である。

【図２１】本発明の第１の実施形態における、第2の再エンコード方法を示す説明図である。

【図２２】本発明の第１の実施形態における、つなぎ目のオーディオ・ビデオ間の時間的不整合を示す説明図である。

【図２３】本発明の第１の実施形態における、非破壊編集結果の第1の管理方法を示す説明図である。

【図２４】本発明の第１の実施形態における、非破壊編集結果の第２の管理方法を示す説明図である。

【図２５】本発明の第１の実施形態における、再生処理時の第1の読み出し順を示す説明図である。

【図２６】本発明の第１の実施形態における、再生処理時の第２の読み出し順を示す説明図である。

【図２７】本発明の第２の実施形態における、非破壊編集結果の管理方法を示す説明図である。

【図２８】本発明の第３の実施形態における、アフレコ結果の管理方法を示す説明図である。

【図２９】従来技術における再エンコード方法を示す説明図である。

【図３０】従来技術における非破壊編集結果の管理方法を示す説明図である。

【符号の説明】

100 バス
101 ホストCPU
102 RAM
103 ROM
104 ユーザインタフェース
105 システムクロック
106 光ディスク
107 ピックアップ
108 ECCデコーダ
109 ECCエンコーダ
110 再生用バッファ
111 記録/アフレコ用バッファ
112 デマルチプレクサ
113 マルチプレクサ
114 多重化用バッファ
115 オーディオデコーダ
116 ビデオデコーダ

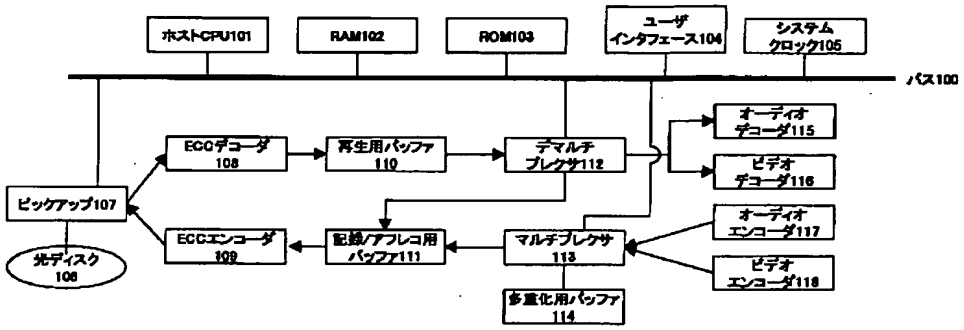
17

117　オーディオエンコーダ

18

118　ビデオエンコーダ

【図１】



【図５】

```
Track header atom {
        Atom size
        Type(='tkhd')
        Version
        Flags
        Creation time
        Modification time
        Track ID
        Reserved
        Duration
        Reserved
        Layer
        Alternate group
        Volume
        Reserved
        Matrix structure
        Track width
        Track height
}
```

【図３】

```
Movie atom {
        Atom size
        Type(='moov')
        Movie header atom
        Track atom (video track)
        Track atom (main audio track)
                :
        User data atom
}
```

【図４】

```
Track atom {
        Atom size
        Type(='trak')
        Track header  atom
        Edit atom
        Track reference atom
        Media atom
        User data atom
                :
}
```

【図６】

```
Media atom {
        Atom size
        Type(='mdia')
        Media header  atom
        Handler reference atom
        Media information atom
        User data atom
                :
}
```

【図７】

```
Media information atom {
        Atom size
        Type(='minf')
        {Video or Sound or Base} media information header atom
        Handler reference atom
        Data information atom
        Sample table atom
}
```

【図８】

```
Sample table atom {
        Atom size
        Type(='stbl')
        Sample description atom
        Time-to-sample atom
        Sync sample atom
        Sample-to-chunk atom
        Sample size atom
        Chunk offset atom
}
```

【図１０】

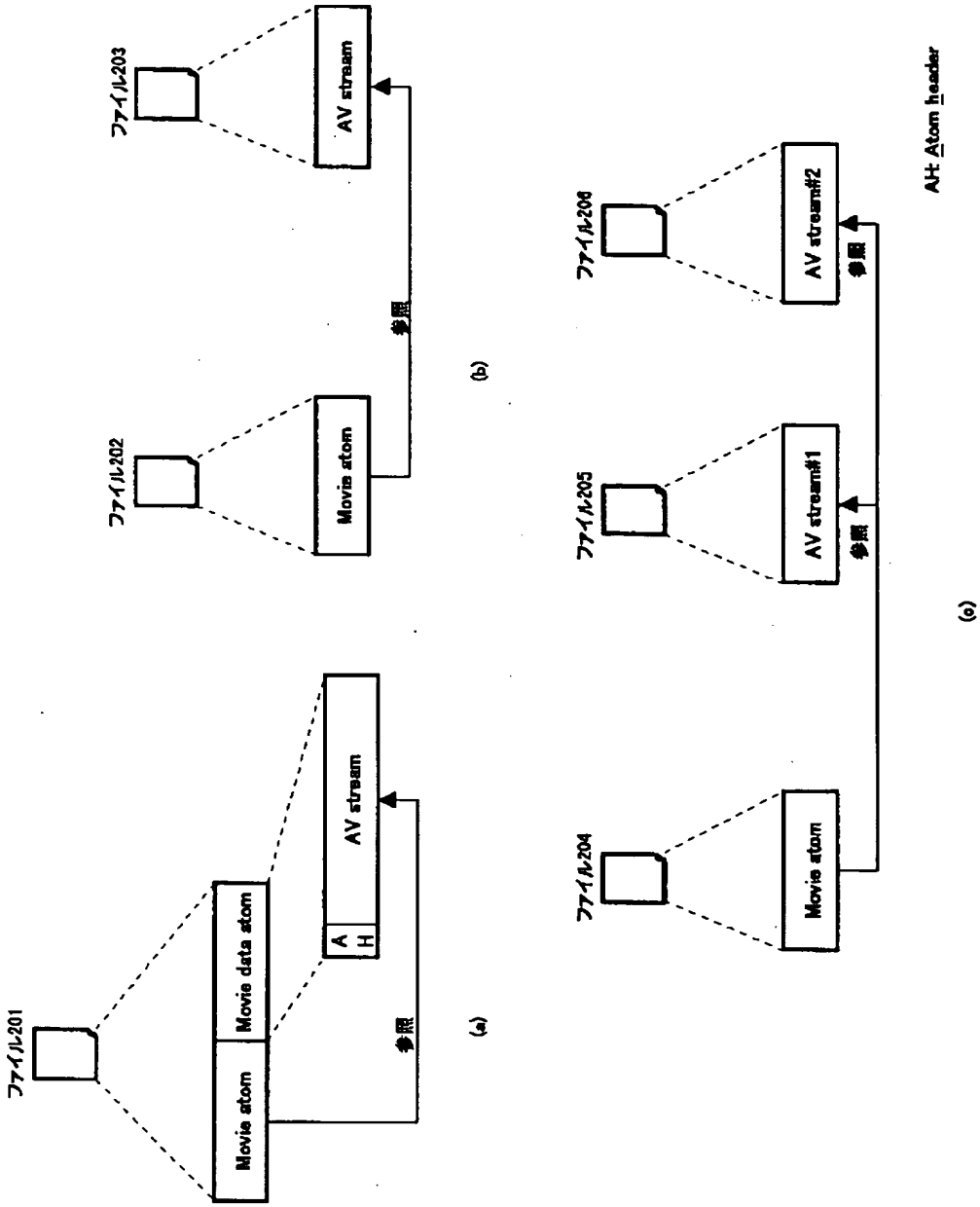
```
Edit atom {
        Atom size
        Type(='edts')
        Edit list atom
}

Edit list atom {
        Atom size
        Type(='elst')
        Versions
        Flags
        Number of entries(=N)
        for (i = 0; i < N; i++){
                Track duration
                Media time
                Media rate
        }
}
```

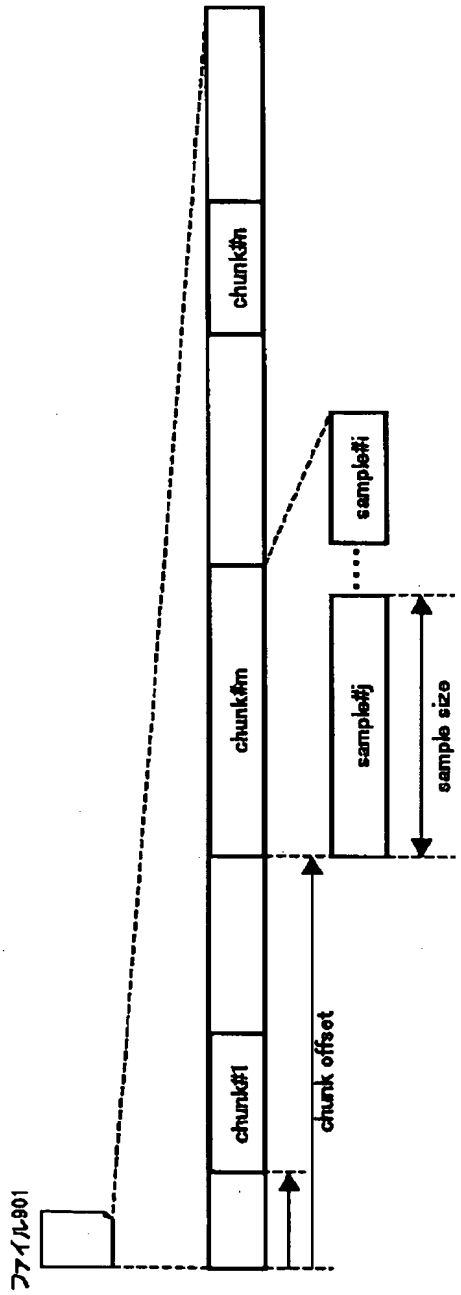
【図１２】

```
User data atom {
        Atom size
        Type(='udta')
        for (i=0;i<N; i++){
                Atom size
                Type
                User data
        }
}
```

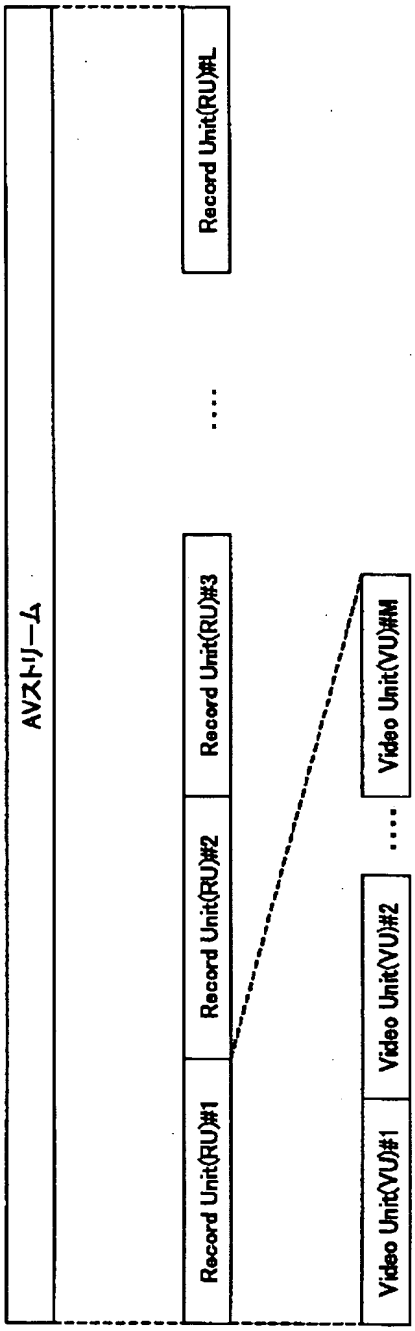
【図２】


ファイル201
Movie atom
Movie data atom
AV stream
A H
参照
(a)
ファイル202
Movie atom
ファイル203
AV stream
参照
(b)
ファイル204
Movie atom
ファイル205
AV stream#1
ファイル206
AV stream#2
参照
A+ Atom header
(c)

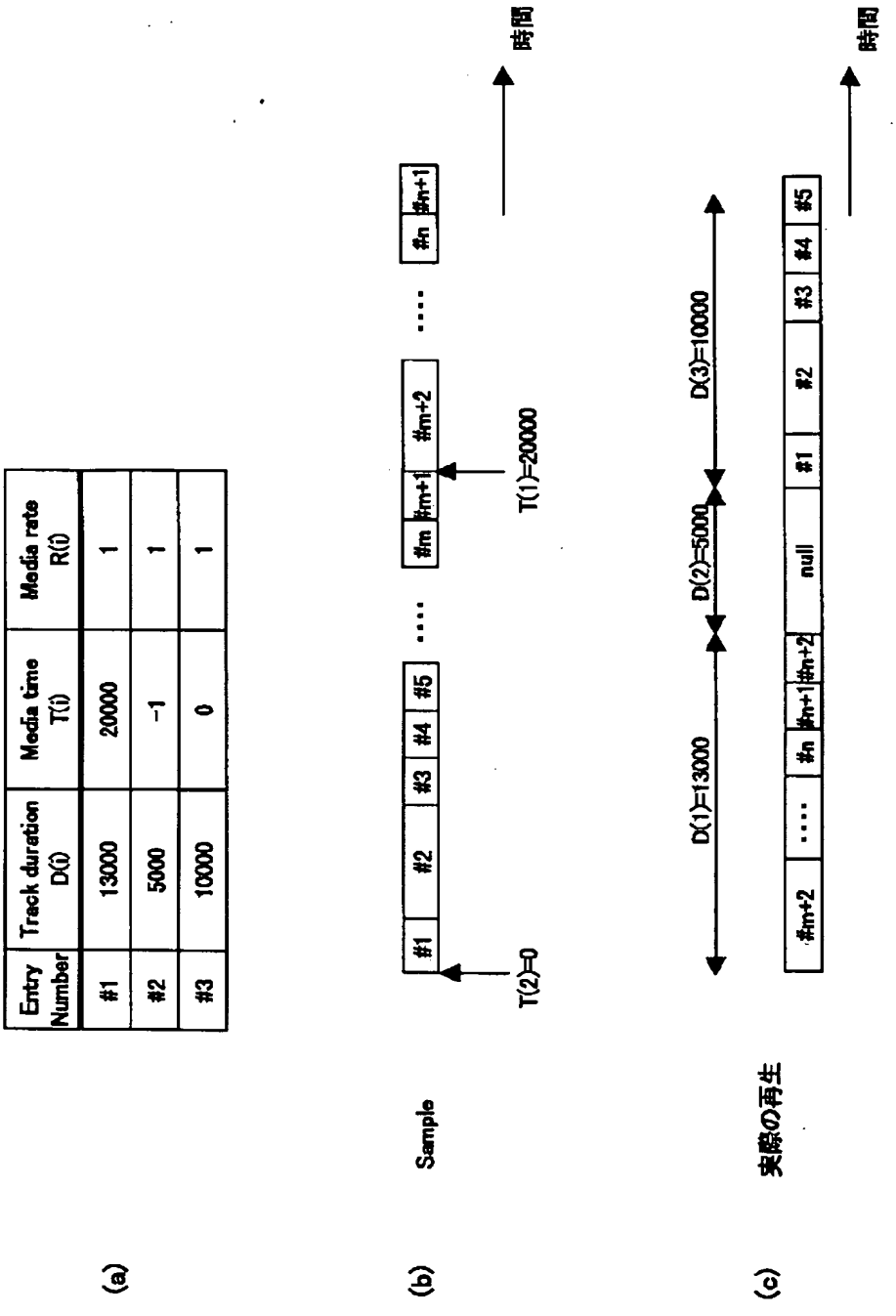
【図９】


ファイル801
chunk#1
chunk#m
chunk#n
chunk offset
sample#j
sample#i
sample size

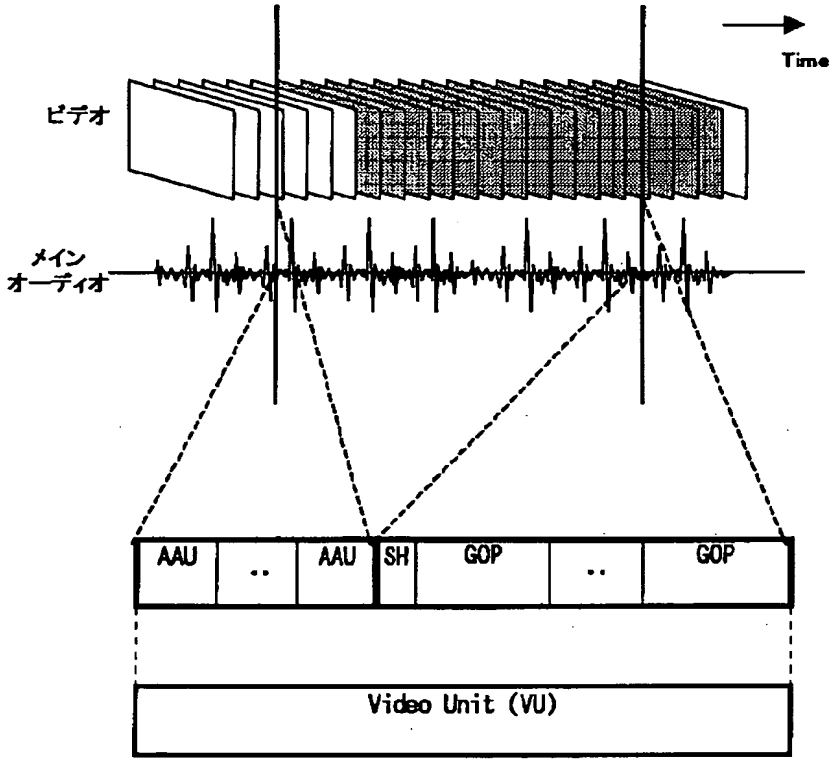
【図１３】


AVストリーム
Record Unit(RU)#1
Record Unit(RU)#2
Record Unit(RU)#3
Record Unit(RU)#L
Video Unit(VU)#1
Video Unit(VU)#2
Video Unit(VU)#M

【図１１】

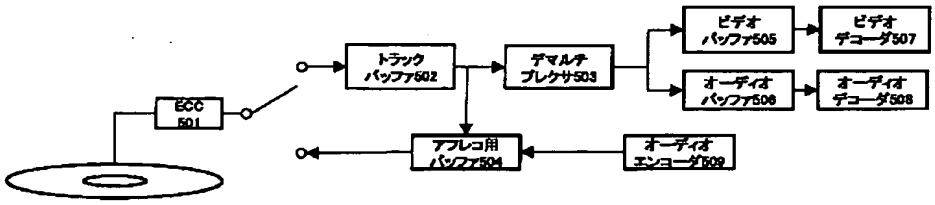

(a)
Entry Number
Track duration D(i)
Media time T(i)
Media rate R(i)
#1
13000
20000
1
#2
5000
-1
1
#3
10000
0
1
(b)
Sample
時間
#1
#2
#3
#4
#5
#m
#m+1
#m+2
#n
#n+1
T(2)=0
T(1)=20000
(c)
実際の再生
時間
D(1)=13000
D(2)=5000
D(3)=10000
#m+2
#n
#n+1
#m+2
null
#1
#2
#3
#4
#5

【図１４】

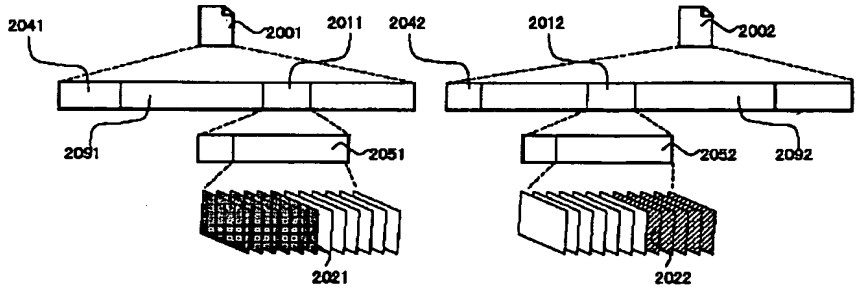


AAU: Audio Access Unit
SH: Sequence Header

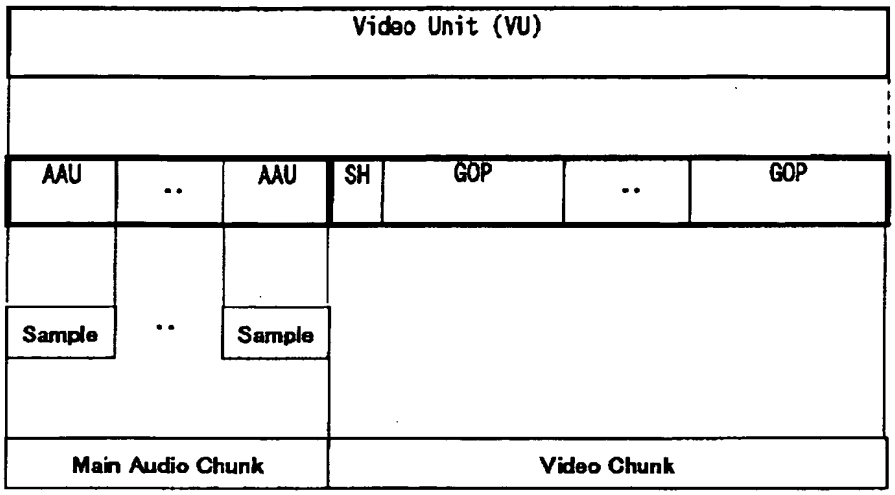
【図１６】



【図１９】



【図１５】

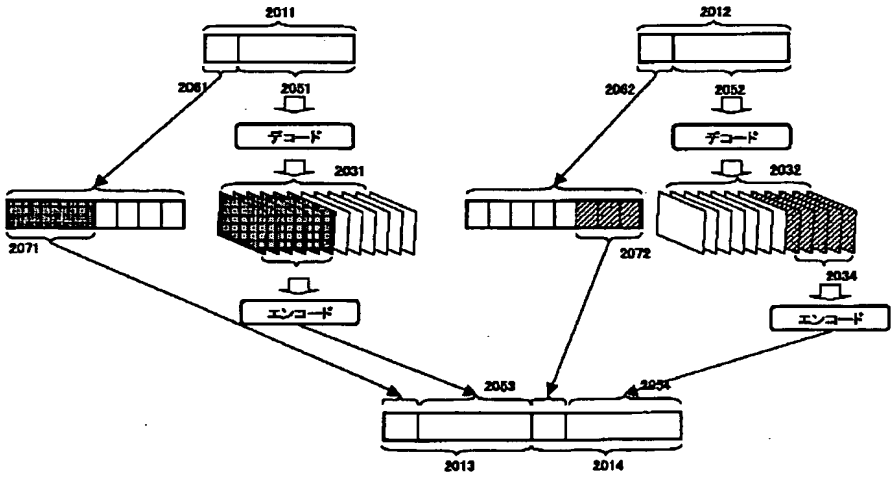


AAU: Audio Access Unit
GOP: Group Of Pictures
SH: Sequence Header

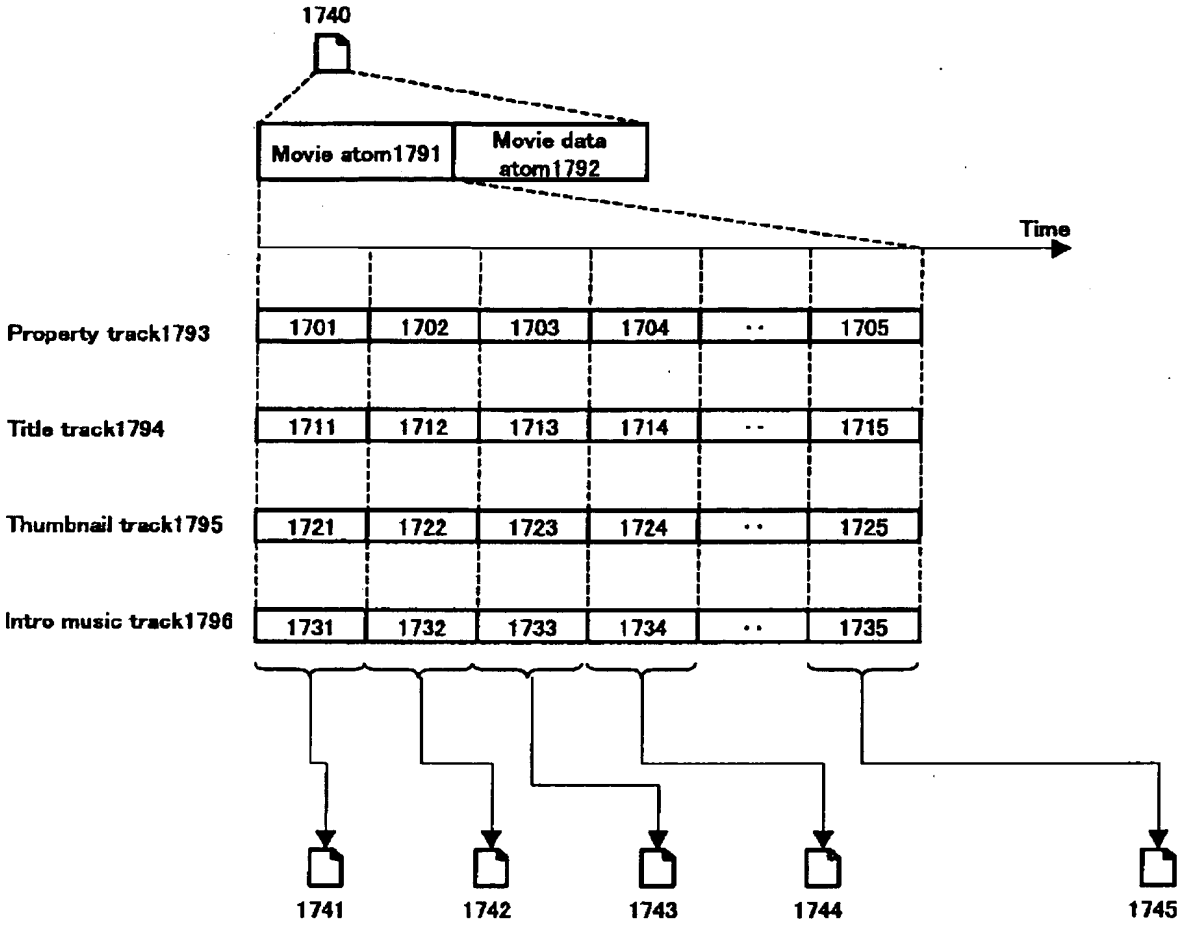
【図１８】

```
Property Entry {
        version
        pe-flags
        parent-entry-number
        entry-number
        set-dependent-flags
        user-private-flags
        reserved
        creation-time
        modification-time
        duration
        binary-file-identifier
        referred-counter
        referring file list
        URL file identifier
}
```

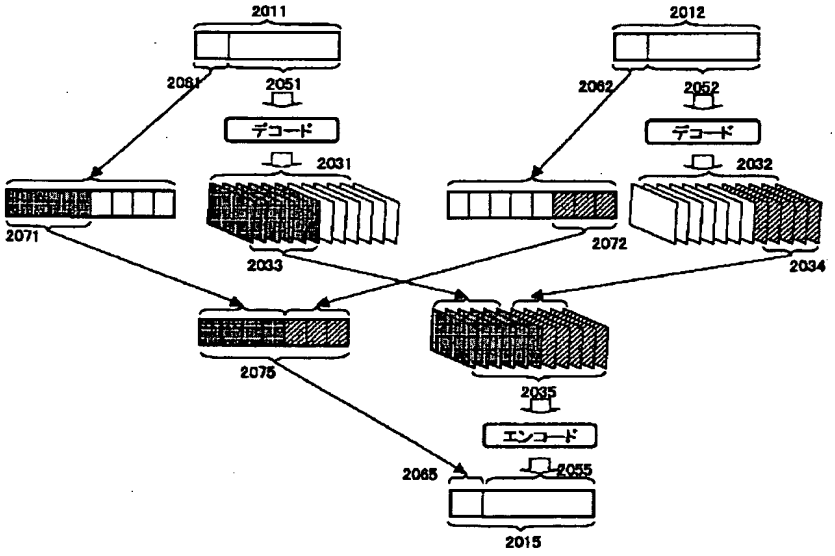
【図２０】



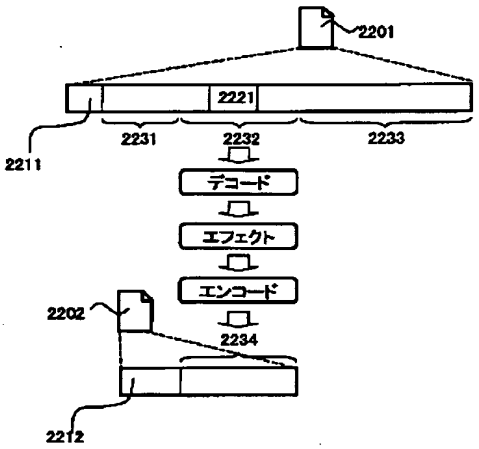
【図１７】


1740
Movie atom1791
Movie data atom1792
Time
Property track1793
Title track1794
Thumbnail track1795
Intro music track1796
1701
1702
1703
1704
1705
1711
1712
1713
1714
1715
1721
1722
1723
1724
1725
1731
1732
1733
1734
1735
1741
1742
1743
1744
1745

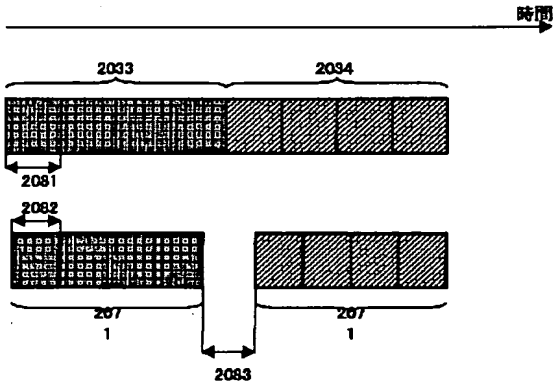
【図２１】


2011
2012
2081
2051
2062
2052
デコード
デコード
2031
2032
2071
2072
2033
2034
2075
2035
エンコード
2065
2055
2015

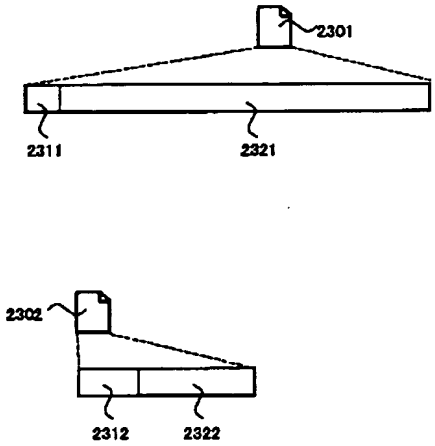
【図２７】


2201
2221
2231
2232
2233
2211
デコード
エフェクト
エンコード
2202
2234
2212

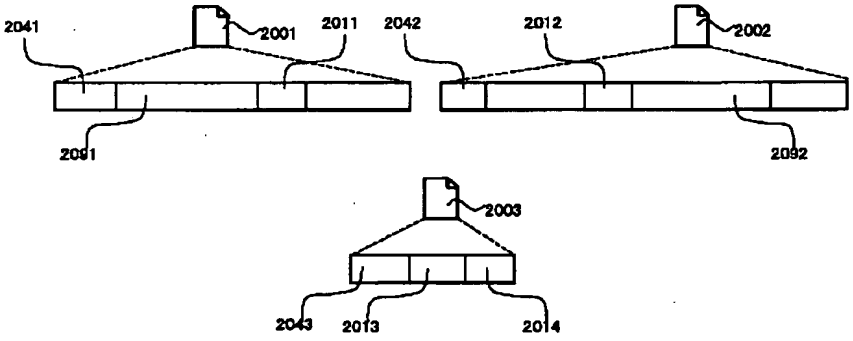
【図２２】



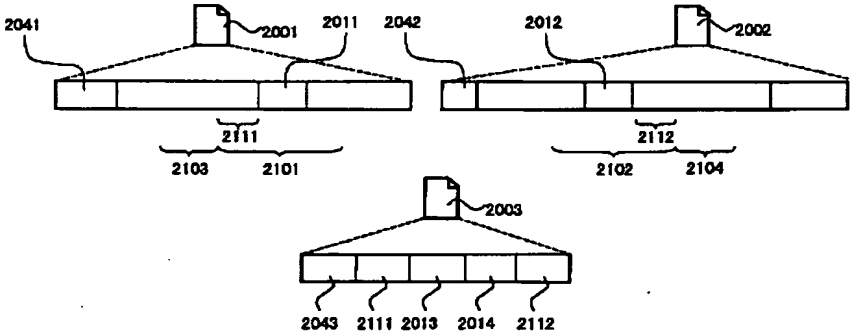
【図２８】



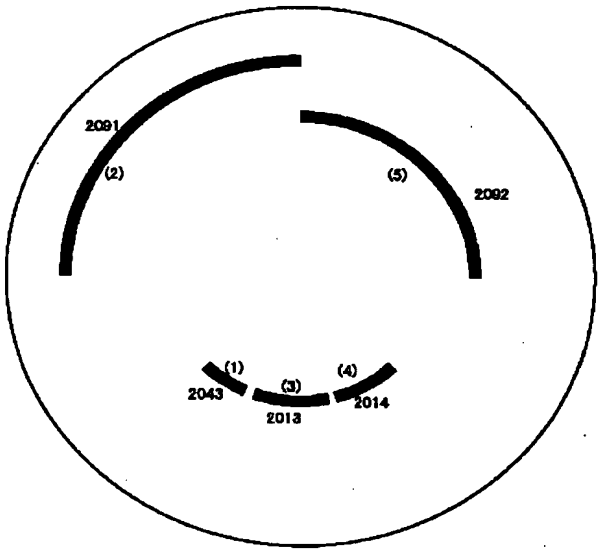
【図２３】



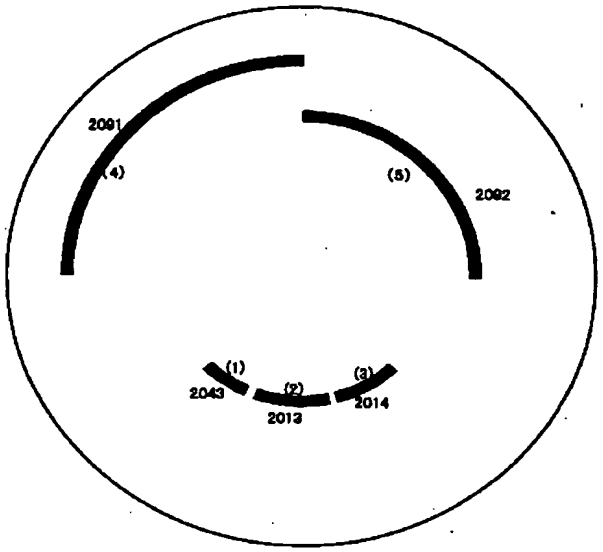
【図２４】



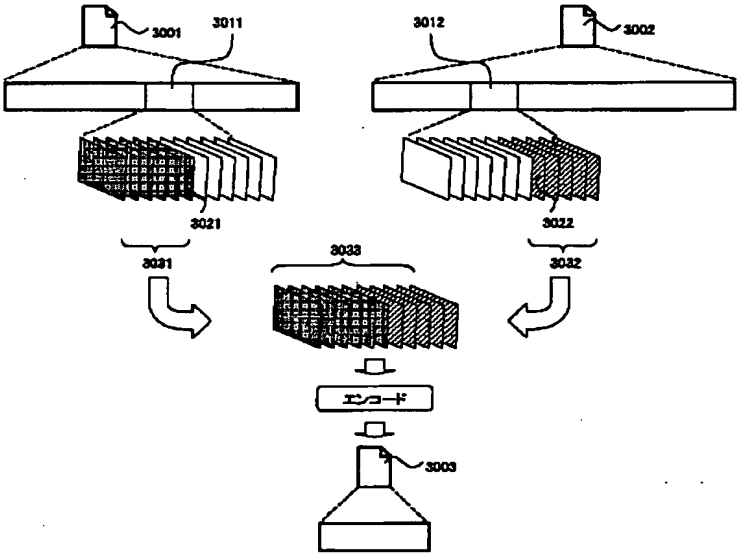
【図２５】


2091
(2)
(5)
2092
(1)
(3)
(4)
2043
2013
2014

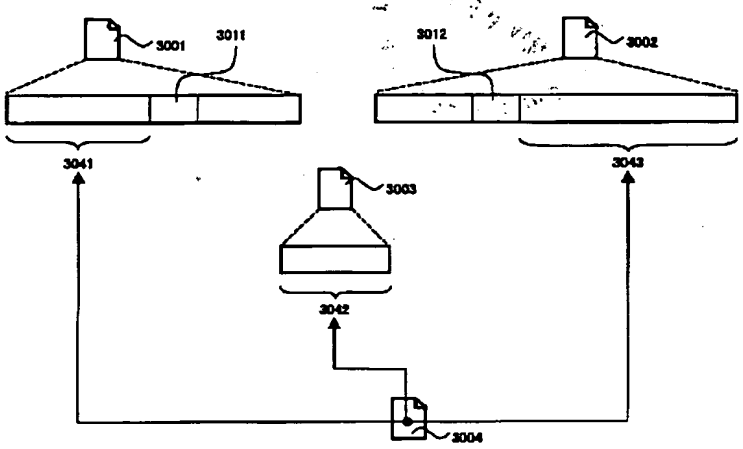
【図２６】


2091
(4)
(5)
2092
(1)
(2)
(3)
2043
2013
2014

【図２９】


3001
3011
3012
3002
3021
3022
3031
3032
3033
エンコード
3003

【図３０】



フロントページの続き

(72)発明者　山口　孝好
大阪府大阪市阿倍野区長池町22番22号　シャープ株式会社内

Ｆターム(参考)　5C053 FA14 FA23 GA11 GB15 GB37
5D044 AB07 BC04 CC06 DE32 DE48
GK12 GM21 HL16
5D110 AA14 AA29 BB01 BB20 CA05
CA06 CA31 CB08 CC06 CF21
DA12 DB03 DC05 DC16 DE01